Building Consistent and Highly Available Distributed Systems
with Apache Ignite and GridGain

Valentin Kulichenko

GridGain Lead Architect & Apache Ignite PMC
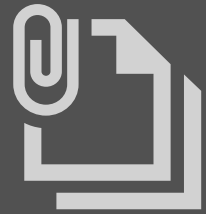
Agenda

- What is Ignite/GridGain?
- Ignite/GridGain and CAP Theorem
- High Availability in Ignite/GridGain
    - Cache Backups
    - Persistence
    - Data Center Replication
    - Data Snapshots (+Demo)
- Q&A

the **in-memory** computing platform
that is **durable**, **strongly consistent** and **highly available**
with powerful **SQL**, **key-value** and **processing** APIs

GridGain In-Memory Computing Platform
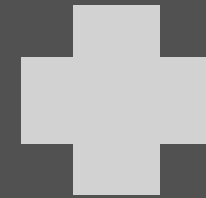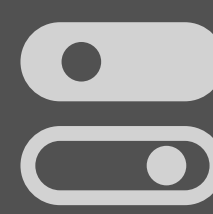

Financial Services

Telco

Travel & Logistics

E-Commerce

Pharma & Healthcare

IoT

SQL

Key/Value

Transactions

Compute

Services

Streaming

ML

Memory-Centric Storage

Ignite Native Persistence
(Flash, SSD, Intel 3D XPoint)

Third-Party Persistence
(RDBMS, HDFS, NoSQL)
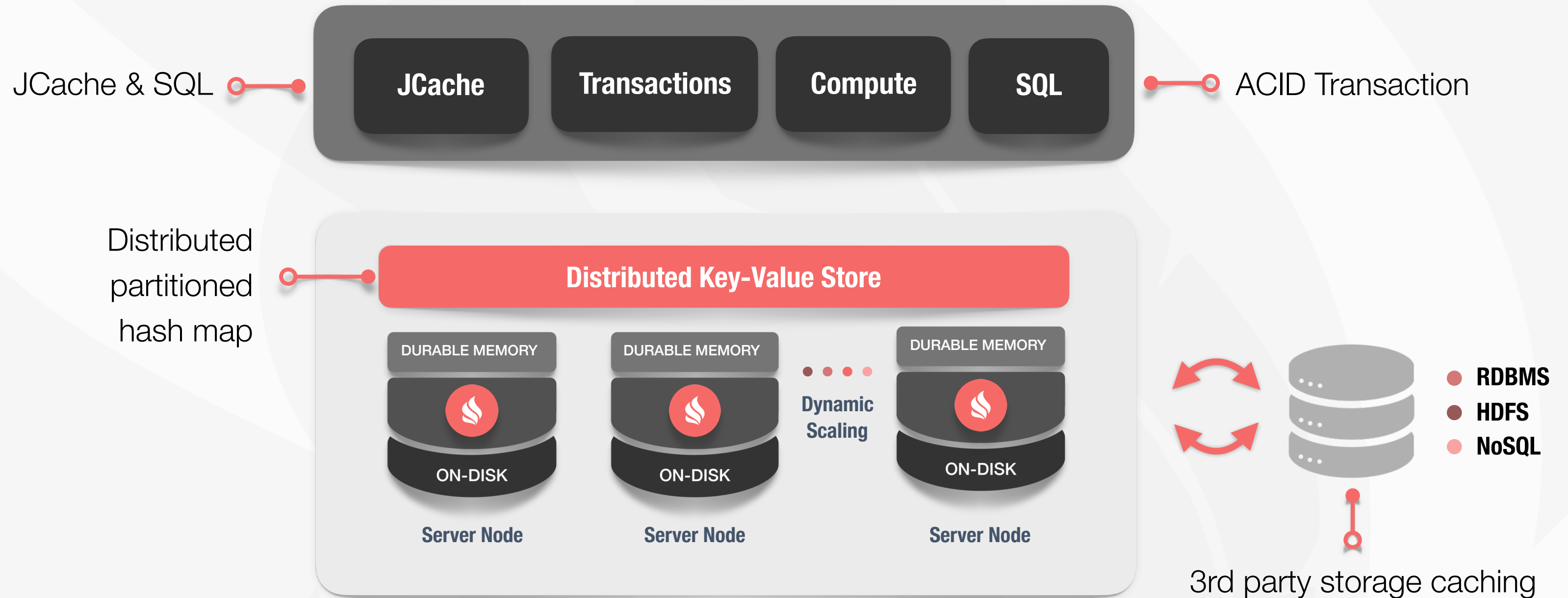
Data Snapshots & Recovery

Monitoring & Management

Security & Auditing

Data Center Replication

Data Grid

JCache & SQL

JCache | Transactions | Compute | SQL

ACID Transaction

Distributed
partitioned
hash map

Distributed Key-Value Store

DURABLE MEMORY

ON-DISK

Server Node

DURABLE MEMORY

ON-DISK

Server Node

Dynamic
Scaling

DURABLE MEMORY

ON-DISK

Server Node

- **RDBMS**
- **HDFS**
- **NoSQL**

3rd party storage caching

Distributed SQL

Cross-platform Compatibility

| Java | .NET | C++ | BI | Tools |

DDL, DML Support

| JDBC | ODBC | SQL API |

SELECT, UPDATE, INSERT, MERGE, DELETE, CREATE and ALTER

Apache Ignite Cluster

Indexes in RAM or Disk

Dynamic Scaling

DURABLE MEMORY

DURABLE MEMORY

DURABLE MEMORY

ON-DISK

ON-DISK

ON-DISK

Server Node

Server Node
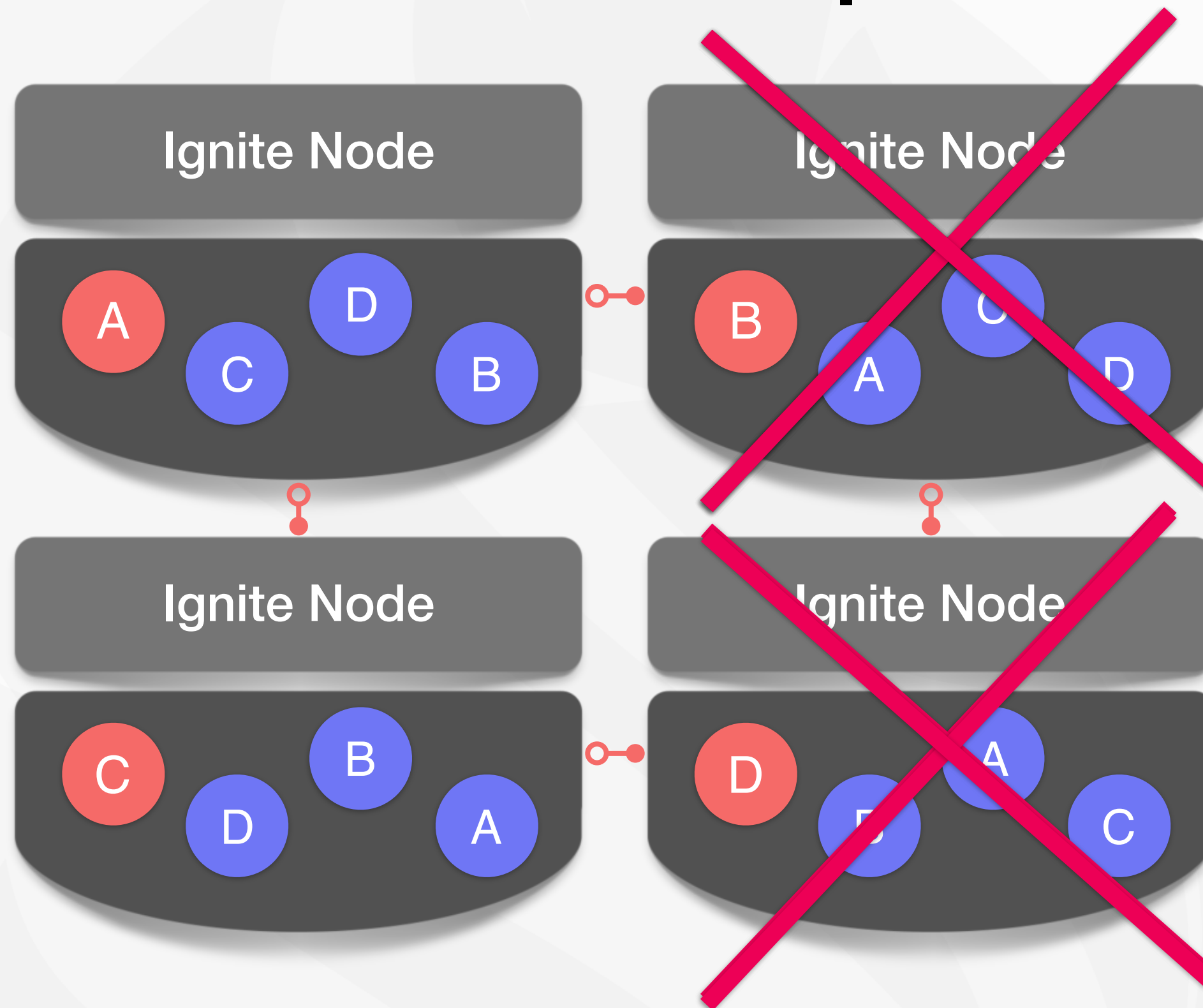
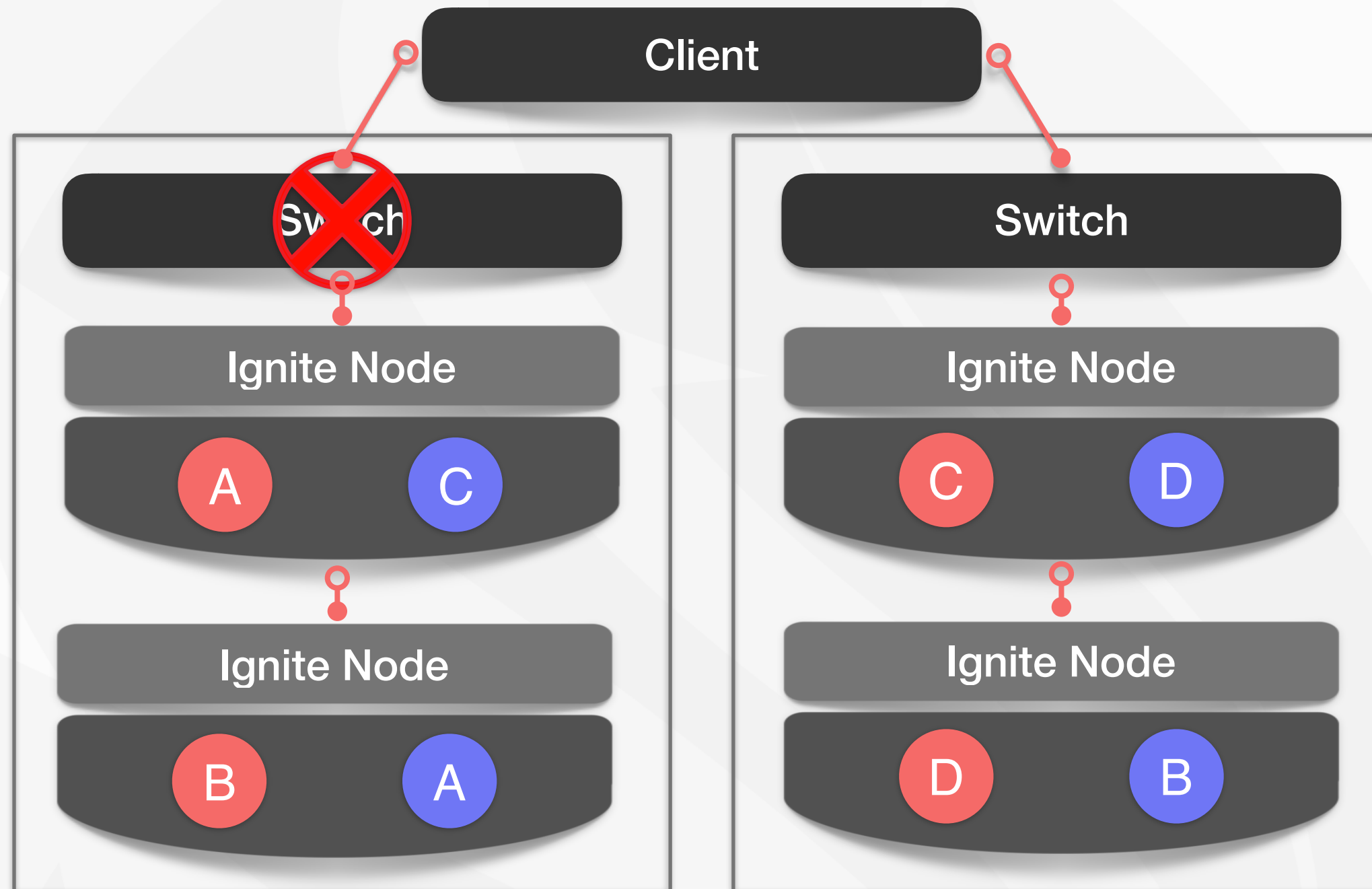Server Node

GridGain and CAP Theorem

- CAP
  - Consistency (C)
  - Availability (A)
  - Partition Tolerance (P)
  - Impossible: CAP
  - Most of distributed systems are CP or AP
- GridGain
  - Strongly CP
  - But still highly available?

C
A?
P

Cache Backups

Rack Safety: Problem

Partition A is lost!

Rack Safety: Solution

- Assign attribute to each node:

```xml
<property name="userAttributes">
    <map>
        <entry key="rack" value="rack-1"/>
    </map>
</property>
```
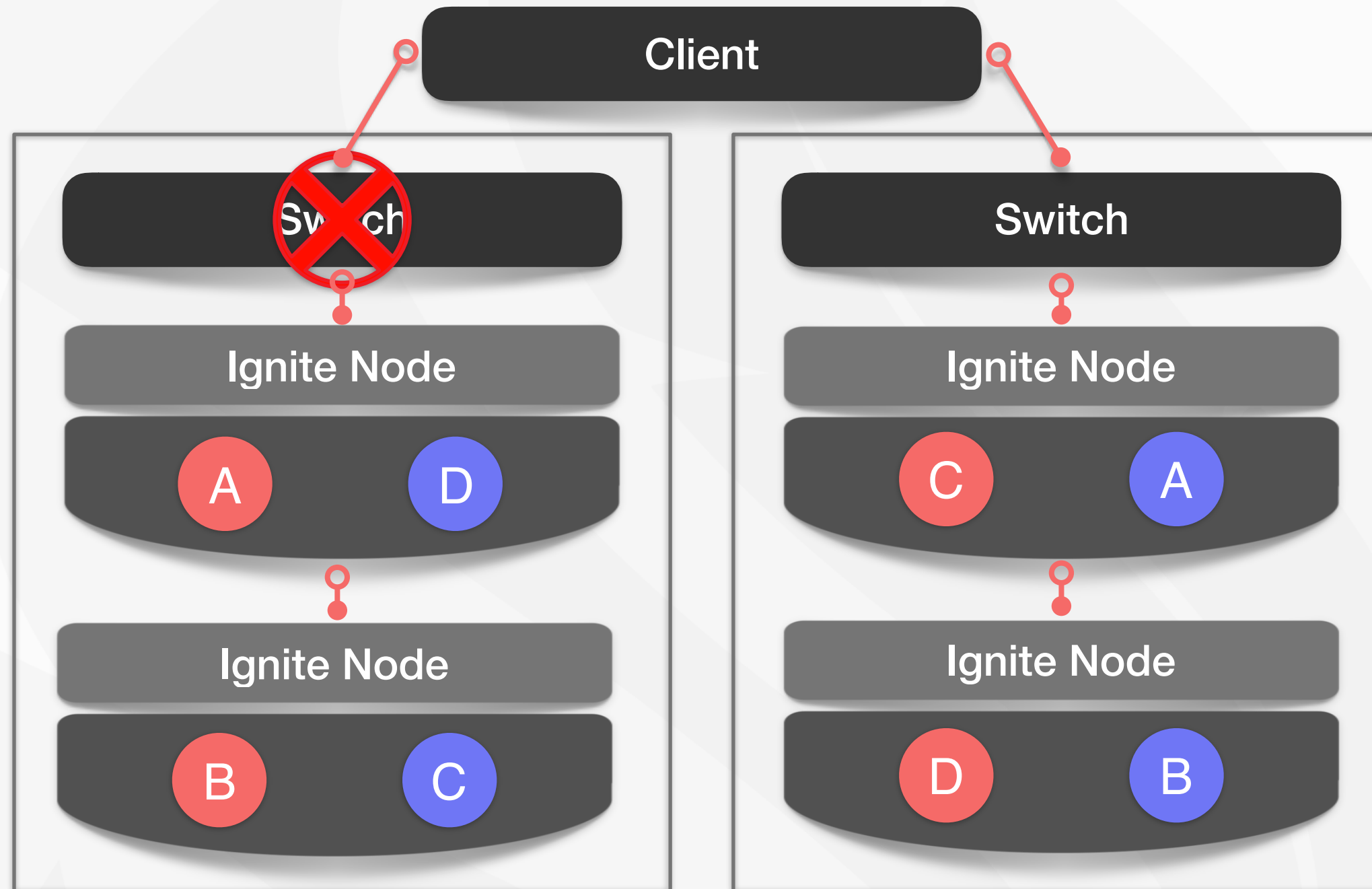
- Implement affinity backup filter:

```java
public class RackFilter implements IgniteBiPredicate<ClusterNode, List<ClusterNode>> {
    @Override public boolean apply(ClusterNode candidate, List<ClusterNode> assigned) {
        String candidateRack = candidate.attribute("rack");
        String primaryRack = assigned.get(0).attribute("rack");

        return !Objects.equals(candidateRack, primaryRack);
    }
}
```
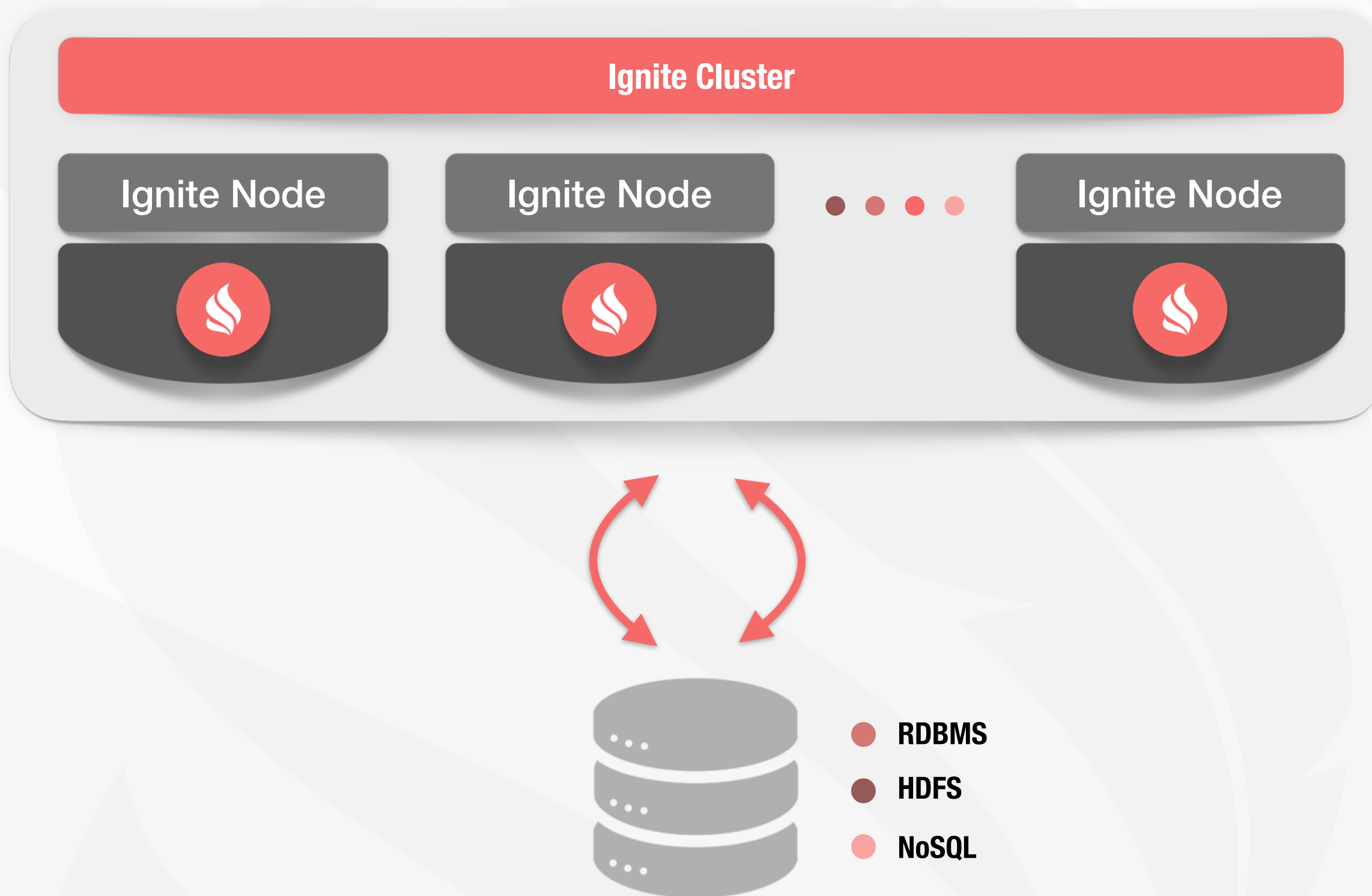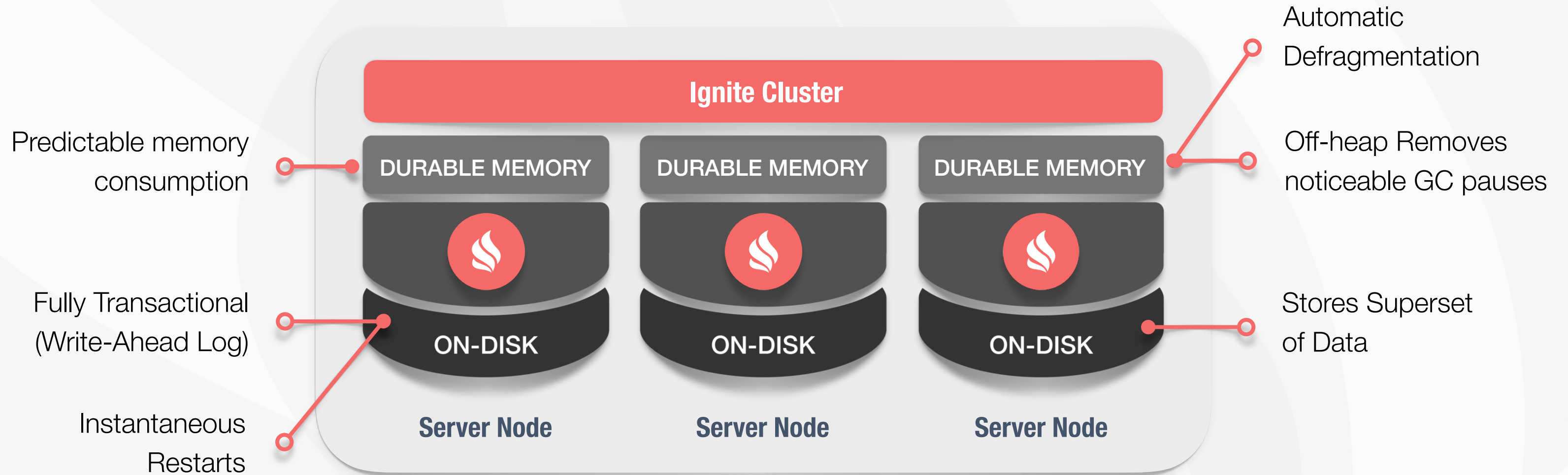
Rack Safety: Solved!

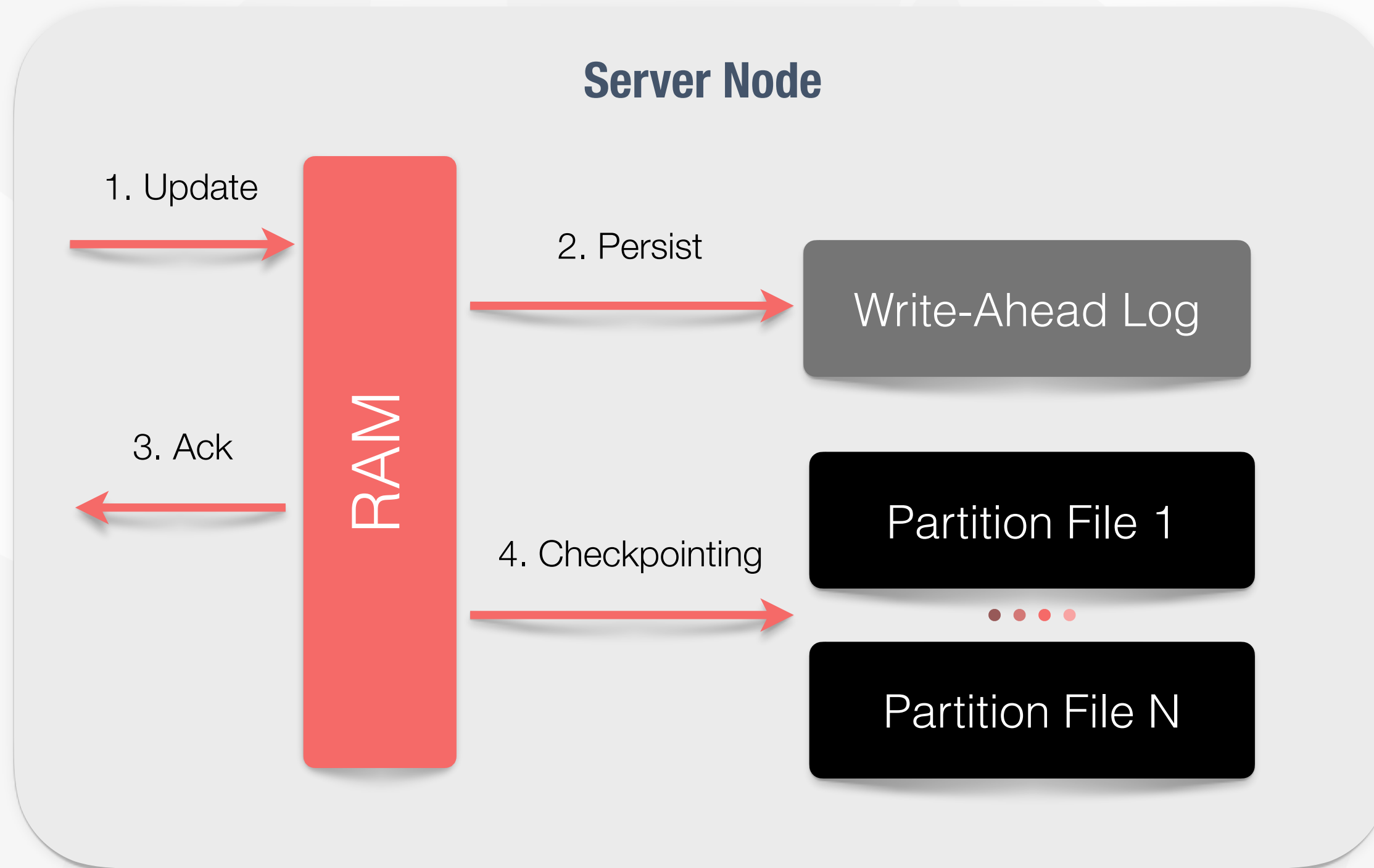# Persistence

Ignite Cluster

Ignite Node Ignite Node Ignite Node

- RDBMS
- HDFS
- NoSQL

Durable Memory

Automatic Defragmentation

Off-heap Removes noticeable GC pauses

Predictable memory consumption

Ignite Cluster

DURABLE MEMORY

DURABLE MEMORY

DURABLE MEMORY

Stores Superset of Data

Fully Transactional (Write-Ahead Log)

ON-DISK

ON-DISK

ON-DISK

Instantaneous Restarts

Server Node

Server Node

Server Node

Ignite Native Persistence

GridGain Data Center Replication

Active-Passive Replication for Disaster Recovery

Active-Active Bi-Directional Replication

Split-Brain Resolution

DC1 DC2 DC3 DC4 DC5

Data Snapshots & Recovery

GridGain Cluster

Data & Indexes

DURABLE MEMORY   DURABLE MEMORY   DURABLE MEMORY

ON-DISK   ON-DISK   ON-DISK

Server Node   Server Node   Server Node

Full & Incremental Backups

Backup

External Store

Restore

Managed Backup Schedules
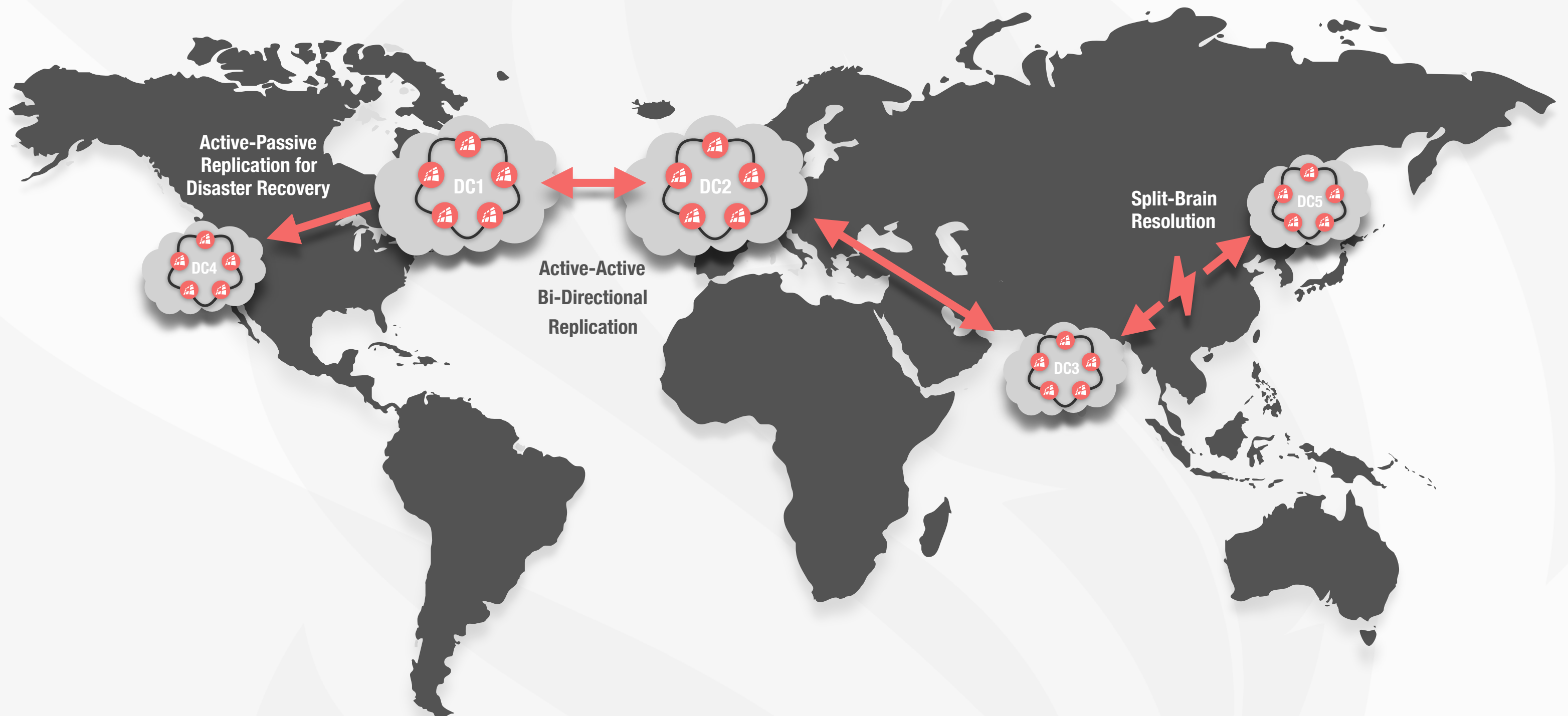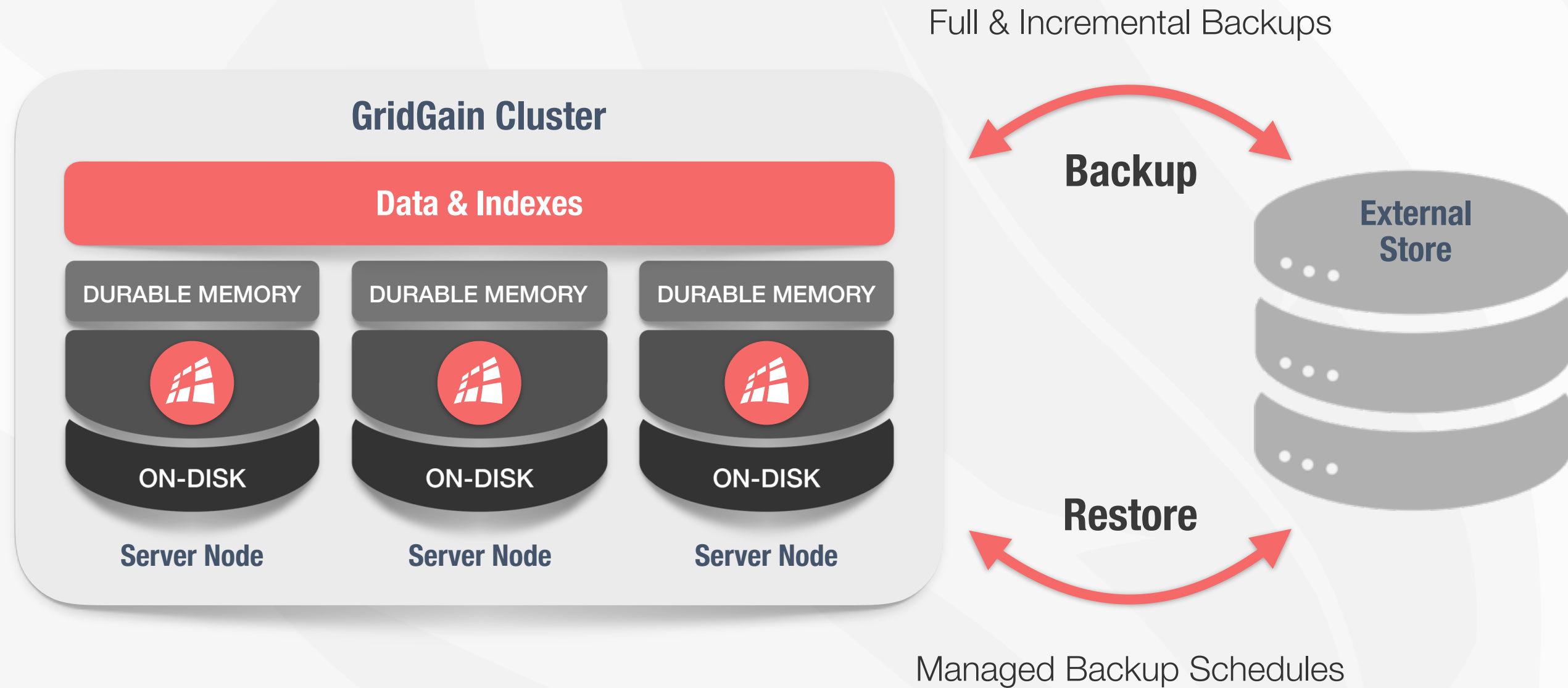
Any Questions?

Thank you for joining us. Follow the conversation.
http://ignite.apache.org

@apacheignite
@vkulichenko